International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

Comparison of Convolutional Neural Networks Model Using Different Optimizers for Image Classification

I Ketut Adi Wirayasa[a]*, Handri Santoso[b], Eko Indrajit[c]

[a,b,c]*Universitas Pradita, Scientia Business Park Tower I, Jl. Boulevard Gading Serpong Blok O/1, Summarecon - Serpong, Tangerang, Indonesia*

[a]*Email: ketut.adi.wirayasa@student.pradita.ac.id,* [b]*Email: handri.santoso@pradita.ac.id,*

[c]*Email: eko.indrajit@pradita.ac.id*

Abstract

Face detection technology and image classification are widely used in several industries that help humans in obtaining information and other related matters. In this paper, the utilization of the Computer Vision system uses the Convolutional Neural Network (CNN) algorithm to classify images by distinguishing the gender of the detected object. Architectural model through transfer learning by experimenting with three pre-trained models, namely VGG-16, Inception-V3, and MobileNet-V2 to determine the best architecture by using Optimizer Adam and RMSProp. To produce the best model and performance, experiments were carried out using several modules such as the data augmentation module and the re-indexing module. The Inception-V3 model got the best results in predicting Gender from the image with an accuracy and loss validation value of 0.9350, 0.1550, compared to VGG-16 and MobleNet-V2 with values 0.9320, 0.1660, and 0.8760, 0.3000.

Keywords: Classification; CNN Architecture; Optimizers.

1. Introduction

Object recognition systems such as: faces, animals, cars, audio and visual or self-driving vehicles, as well as many other applications that utilize Computer Vision techniques over the last few years have developed and become a technology that is widely used and adopted in various aspects in human life nowadays [1,3].

---

* Corresponding author.

The increase in the use of Artificial Intelligence-based applications is one that contributes to the increase in the number of jobs and the use of technology related to Computer Vision. One method that is quite well known and widely used today in image classification is using Convolutional Neural Networks (CNN). CNN is a technology that combines Artificial Neural Networks and Deep Learning methods [4]. CNN can provide higher accuracy on dataset types such as images or objects, when compared to other objects such as word datasets or signal waves. This is because the convolution processing on CNN uses a matrix structure in pixel size. Mapping on pixels is structured like a matrix. Therefore, CNN is easier to use on image objects or matrices [5]. CNN is a classification method and belongs to the deep learning group that uses a convolution layer. CNN has two main stages, namely feature learning and classification. The feature learning stage consists of a convolution layer, ReLU (activation function) and a pooling layer. While the classification stage consists of a flatten, fully-connected layer, and prediction as well as each part has two main processes, namely feedforward and backpropagation[6]. RMSProp is a gradient-based optimization technique used in training neural networks. RMSProp was developed as a stochastic technique for mini-batch learning. RMSProp uses adaptive learning speed rather than treating learning speed as a hyper parameter [7,8]. Adam (Adaptive moments) is a gradient-based optimization algorithm from a stochastic function where the first moment is normalized by the second moment and gives the direction of the update. The update operation considers only the smooth gradient version and also includes a bias correction mechanism [7,9]. This study will complementing these previous works and researches with provide evaluation and sample of various image on deep learning based face classification performance with the goal of this work is to provide some answers by research questions such as: Does image quality affect to the image classification by pre-trained models? How should image classifier be computed?

The following contributions in this paper:

- The studying an empirically evaluate of image and the characteristics on the image classification performance of three deep CNN models on the image dataset.
- Provide the comparative evaluation of the three deep CNN models, namely, VGG-16, MobileNet-V2 and Inception-V3 with potential areas for the improvement.

2. Related Works

Convolutional Neural Network (CNN) or ConvNet is one of the Deep Learning algorithms that is widely used in receiving input data in the form of images or objects that are used by machines to do learning in recognizing images or objects and distinguish one image from another, and one object to another. CNN is a method developed from Multi-Layer Perceptron's (MLP) and has more dimensions than MLP. CNN has input arrays ranging from two-dimensional to more and in deeper learning, convolution neural networks are the most commonly applied to analyze visual images [1,3,10]. CNN is a powerful deep learning algorithm and specialized in image processing, capable of dealing with millions of parameters and can converting the original 1D signal into the 2D signal with widely technique to apply in other applications [2,11]. There are some note for CNN, it does not require for hand-crafted feature extraction. CNN architectures do not necessarily require segmentation of some image, CNN need more data because of its millions of learnable parameters to estimate, and more computationally expensive, resulting in requiring Graphical Processing Units (GPU) for model

training to accelerate training speed [2,11,12]. By having a high level of accuracy, the number of feature extractions generated by convolution and using relatively little pre-processing compared to other image classification algorithms, CNN is widely used in image or object classification. By having a function to perform feature extraction. These features need to be obtained to perform processes or tasks such as classification, clustering or regression. By using CNN, the feature extraction process is carried out automatically at the Convolutional layer, Pooling layer and also the activation of the Rectified Linear Unit (ReLU) for further classification of these features in the Fully Connected layer (FCL) and Softmax activation [13]. According to [14], to compare the quality of different models, the existing models were collected and analyzed for their accuracy values based on the results obtained by previous researchers, and the results can be seen as in Figure 1. From these results it was found that with different sampling techniques, it is not possible to do a direct comparison by utilizing resource data used with different specifications.

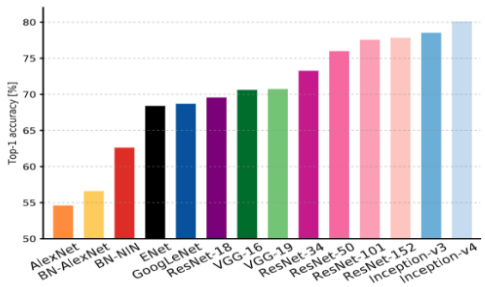

Figure 1: Comparison of Accuracy vs. Model [14]

By looking at the comparison of accuracy and model selection that has been done by previous researchers. The image classification models that will be selected for gender detection are the GoogleNet/Inception-V3, MobileNet-V2 and VGG-16 models. The results of the comparison to this image will be compared and their performance analyzed.

3. Model Architectures and Experiment

Transfer learning is a common and effective strategy to train a model efficiently which can use the existing knowledge learned from one environment and solve the other new problem by using a small amount of dataset and to achieve high accuracy with a short training time [2,15] One way to get optimal performance is by selecting the model and architecture based on the advantages of the model in performing feature extraction. The use of binary indexing can also reduce deployment resources and improve the quality of user interaction with the system. And in the experimental dataset, transfer learning or pre-trained training is used on the CNN architectural model that has been trained using the previous dataset [16].

3.1. Inception-V3

GoogleNet/Inception is a model that focuses on computational costs. The InceptionV3 model is an evolution of GoogleNet (Inception1) which is based on factorized 7x7 convolution and is divided into 2 or 3, 3x3 layer convolution operations with the aim of increasing computational speed to be able to accept images larger than

299x299 pixels [16]. This is because this model only has about 4 million parameters, very small compared to VGG, but with a more complex architecture. This is because this model does not apply a Fully-Connected layer and replaces it with a pooling layer only. These fewer parameters result in a smaller model size and also a faster model calculation process when compared to other models. The weakness of this model is that when performing computations it requires quite a lot of memory because this model focuses on the width of the convolution layer [1]. The transfer learning performs for image classification based on dataset better than the model based on original deep CNN with tuning the Inception-v3 model can effectively improve the accuracy of the image classiffier [17].

3.2. MobileNet-V2

MobileNet-V2 is a very light and slim architectural model compared to previous models, designed for mobile applications and embedded vision. This model is based on a streamlined architecture that uses depth-MobileNet-V2 by using deep separable convolutions to build lightweight deep neural networks with only 54 layers and an input image size of 224x224 to perform computations quickly [16]. MobileNets uses deep separable convolutions, and has similarities to the Inception-v3 model. This model can reduce the number of parameters and latency. In addition, MobileNets also has a useful shrinkage-parameter model that can be used before the training process is carried out to make it the right size and fit [18,19]. Result study for MobileNet-V2 by comparing two datasets and analysis the experimental found that MobileNet-V2 achieves higher accuracy than MobileNet-V1 model by using the inverted design [20].

3.3. VGG-16

VGG is one of the models that is very simple and easy to understand and is a model that is often used for machine learning and the advantages of this model, VGGNet has a very deep network that includes 16 or more CONV/FC layers, each of which has a convolution layer with small size of 3x3, and interspersed by layers of POOL (where each group consists of 2 or 3 layers of CONV) [16]. Model VGG-16 has good accuracy when using the gradient descent optimization comparing without optimization [21].

3.4. Experiment

a) Dataset sourced from Kaggle

Figure 2: Example of Image Dataset

The secondary dataset used is sourced from Kaggle which consists of approximately 200K facial images and is divided into 2 (two) genders, male and female. By dividing the data into training, testing and validation data with random distribution. Training data will divide into 3 parts due to the specification of the computer specification and we did the pre-trained architecture and combine with the training model from scratch.

b) The tools used are Jupyter Notebook, Tensor flow, and Keras as well as using the Python programming language. Laptop specification Intel Core i5-421oU, 1.70GHz, RAM 8GB, Windows 8.1.

c) Research method propose for experiment:

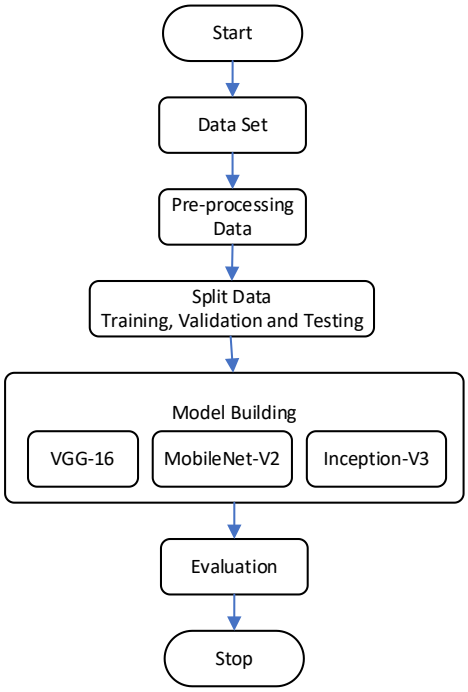

Figure 3: Research Method

d) Architecture system and model created

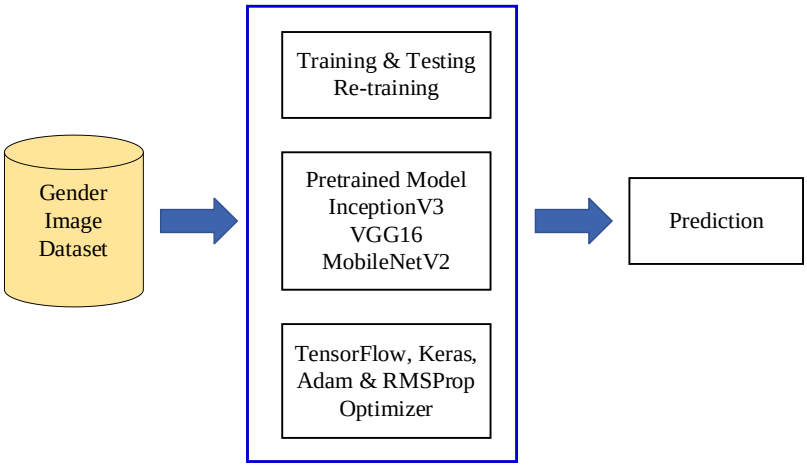

Figure 4: System Architecture

e) Hyperparameter configuration for the training model uses several hyperparameters and applied for all model, such as:

- Epoch: 10-30
- Steps per epoch : 10
- Validation steps : 20
- Learning rate (α): 0.0001 – 0.001
- Optimizer: RMSProp and Adam
- Fine Tuning Layer: 5-30

Where each model is trained with these hyper parameters, and the best results are taken for later comparison and analysis of the performance of each model.

f) Preprocessing is done on the dataset. Before being used for model training, such as cleaning the dataset so that all images fit into their category, after that, manual labeling of all data is carried out. The next step is to perform data augmentation to get a dataset with a larger and more varied size.

g) The experimental stages are carried out as follow:

- Data Preparation: Data is downloaded (extraction of training dataset) on the website kaggle.com (Gender Classification 200K image dataset).
- Dataset for training (training of CNN model) is divide into 3 parts, part one using 50%, 80% and 100% from total actual dataset.
- Pre-processing and data normalization: resize the patch size to 255x255 pixels.
- Determine the type and number of layers used in the CNN architecture. The type and number of layers used in this study for training data are shown in Figure 5.

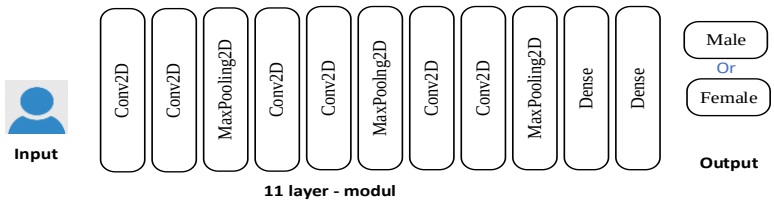

Figure 5: Network layer architecture – that is used.

- Determine the number of patches used for the training process. If the number of images in the dataset is greater than the number of training data used, the system will select a random patch to use for the training process.
- Extraction of features from training datasets. Feature extraction is found in the activation process by determining the ReLU activation layer used for feature extraction. ReLU is on the 7th layer of the CNN architecture used.
- Determine the amount of data in the testing process. Perform feature extraction on dataset testing. The

feature extraction process for testing datasets is similar to feature extraction from training datasets.

- In addition to using transfer learning, a self-built layer model is also made. The use of re-indexing and augmentation modules for image classification using the selected CNN architecture and model.
- To get accuracy results with a loss rate, it is better to compare the CNN model and use modules that can improve the performance of the classification model in addition to the main module, namely the data augmentation module and also the re-indexing module. The data augmentation module is very helpful, especially if the dataset you have is small, while the re-indexing module will be very helpful in the image search process and comparing features between query data features and data features in the database.
- Model training from scratch requires a large dataset (about 10-25k data per class) in order to produce good models. Meanwhile, transfer learning requires a smaller dataset when compared to model training from the start.
- Transfer learning results tend to be better than modeling from scratch, especially the smaller the dataset used. The next development that can be done to improve the performance of this system is to parallelism the model, so that it can extract features from the dataset simultaneously and speed up database creation and addition of data to the database itself.
- The training process requires quite a lot of data. The selection of the right preprocessing method will make the model able to do training faster and increase the final accuracy of the model. Feature learning is carried out in model training to increase the final accuracy of the model.

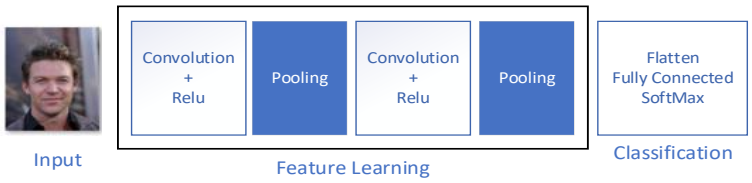

Figure 6: CNN Architecture

- To overcome the problem of overfitting during the model training process, the hyper parameter process is carried out repeatedly until it gets the best hyper parameter value.
- To reduce the overfitting we obtain more training data and we included regularization with dropout or weight decay, batch normalization, and data augmentation, as well as reducing architectural complexity.

4. Result and Analysis

- The training result for the comparison of model is:

Table 1: Performance Model & Architectures – Training

Model & Architectures	Accuracy (%)	Loss (%)
VGG-16	91.00	18.50
Inception-V3	89.90	21.05
MobileNet-V2	87.60	29.70

- The training results on the performance of MobileNet-V2 model is the loss are quite high: 30.00%, and accuracy is 87.60% can be seen as follows:

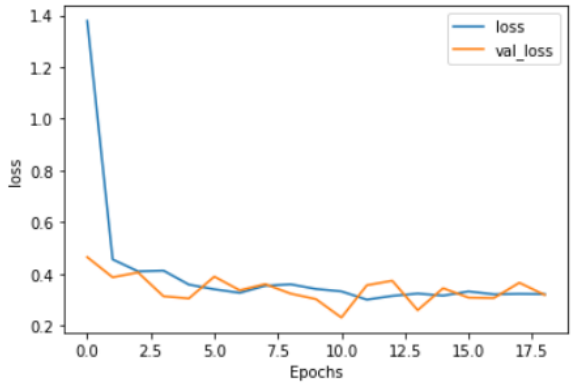

Figure 7: Training & Validation Loss

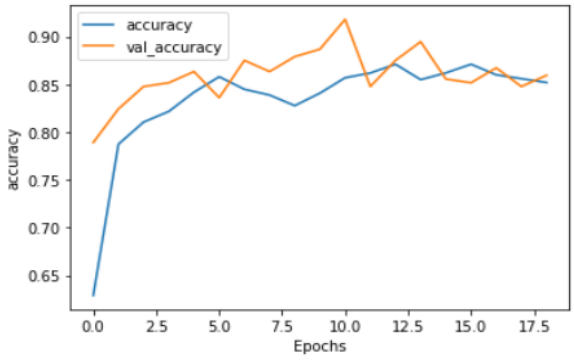

Figure 8: Training & Validation Accuracy

- From the table above, the experimental results show an accuracy of 85-100% with validation loss of 0-30.00% in the third experiment.
- Over fitting happens with refers to regularities specific to the training set situation where a model learns statistical and ends up memorizing the irrelevant noise instead of learning the signal of dataset. This is one of the main challenges in machine learning, as an over fitted model is not generalizable to never-seen-before data. By doing routine check for recognizing over fitting to the training data is to monitor the loss and accuracy on the training and validation set.
- The testing result shown as:

Table 2: Performance Model & Architectures – Testing

Model & Architectures	Accuracy (%)	Loss (%)
VGG-16	93.20	16.60
Inception-V3	93.50	15.50
MobileNet-V2	87.60	30.00

- Each of the above models has its own advantages and disadvantages, for example the lightest model is MobileNet, the fastest and most efficient model is GoogleNet/Inception, and the model with the best accuracy, and loss is VGG16. The shortcomings of these models also vary, ranging from memory, loss value, recall, length of the training process, and the size of the resulting model [22]–[24]. From these strengths and weaknesses, testing experimental results are obtained where the highest accuracy value is 93.50%, the lowest loss value is 15.50%, and the processing time is average 35 ms/step.

Figure 9: Training & Validation – Loss

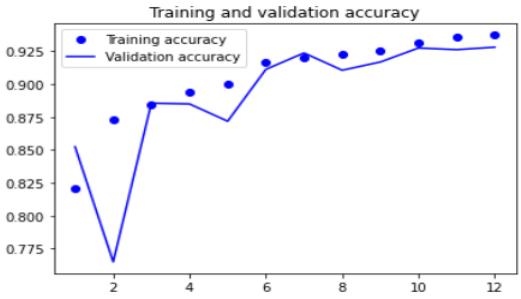

Figure 10: Training & Validation – Accuracy

5. Conclusion

The experimental results show that CNN VGG16, MobileNet-V2 and InceptionV3 models using transfer learning three hidden layers, where each hidden layer consists of a convolutional layer, ReLU activation and max-pooling can classify images of gender - male and female with a good level of accuracy. This is also due to the performance optimizer which is combined during training and data testing using Adam and RMSProp optimizers. The results of testing and evaluation using the RMSProp Optimizer resulted in the lowest loss value for the InceptionV3 model of 0.1550, and also, the highest accuracy value 0.9350. With the results of this

experiment, the conclusion that can be drawn is that the InceptionV3 model has the highest level of accuracy in the training and testing process, using RMSProp optimizer.

6. Limitation and Recommendation

The number of epoch in this study is limited to 30. We did not compare models using different numbers of iterations so that further study could attempt to build a model using a different number of iterations. An in-depth study of the outliers that exist in the data also has not been carried out in this study. The other combination of iterations and robust methods can be carried out in further research. Furthermore, the gender image classification of the Kaggle dataset used through the combination of the performance optimizer can be considered for further research in producing the optimal level of prediction and accuracy from the comparison of the CNN model. The other things is CNN needs more data to get the better result and it's more computationally expensive, resulting in requiring graphical processing units (GPUs) for model training.

References

[1]. M. S. Islam, F. A. Foysal, N. Neehal, E. Karim, and S. A. Hossain, "IncePTB: A CNN based classification approach for recognizing traditional Bengali games," Procedia Comput. Sci., vol. 143, pp. 595–602, 2018, doi: 10.1016/j.procs.2018.10.436.

[2]. A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," Smart Innov. Syst. Technol., vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[3]. Z. J. Wang et al., "CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization," IEEE Trans. Vis. Comput. Graph., vol. 27, no. 2, pp. 1396–1406, 2021, doi: 10.1109/TVCG.2020.3030418.

[4]. X. Han and Y. Li, "The Application of Convolution Neural Networks in Handwritten Numeral Recognition," Int. J. database Theory Appl., vol. 8, no. 3, p. 10, 2015, doi: http://dx.doi.org/10.14257/ijdta.2015.8.3.32.

[5]. S. Liu et al., "Matching-CNN Meets KNN : Quasi-Parametric Human Parsing," IEEE Xplore, p. 9, 2015.

[6]. A. Yusuf, R. C. Wihandika, and C. Dewi, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," J-PTIIK, vol. 3, no. 11, p. 11, 2019, [Online]. Available: http://j-ptiik.ub.ac.id.

[7]. P. Bahar, T. Alkhouli, J. Peter, C. J. Brix, and H. Ney, "Empirical Investigation of Optimization Algorithms in Neural Machine Translation," no. 108, pp. 13–25, 2017, doi: 10.1515/pralin-2017-0005.PBML.

[8]. S. Voronov, I. Voronov, and R. Kovalenko, "Comparative analysis of stochastic optimization algorithms for image registration," 2018.

[9]. R. Zaheer, "A Study of the Optimization Algorithms in Deep Learning," 2019 Third Int. Conf. Inven. Syst. Control, no. March, pp. 536–539, 2020, doi: 10.1109/ICISC44355.2019.9036442.

[10]. S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using Ranking-CNN for Age Estimation," Cvpr, pp.

5183–5192, 2017, [Online]. Available: http://arxiv.org/abs/1905.06509.

[11]. R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First Int. Conf. Secur. Cyber Comput. Commun., no. December 2018, pp. 278–282, 2021, doi: 10.1109/ICSCCC.2018.8703316.

[12]. E. R. Jeong, E. S. Lee, J. Joung, and H. Oh, "Convolutional neural network (Cnn)-based frame synchronization method," Appl. Sci., vol. 10, no. 20, pp. 1–11, 2020, doi: 10.3390/app10207267.

[13]. W. Setiawan and R. Rulaningtyas, "Classification of neovascularization using convolutional neural network model," TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 17, no. 1, pp. 463–472, 2019, doi: 10.12928/TELKOMNIKA.v17i1.11604.

[14]. A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv, p. 7, May 2016, [Online]. Available: http://arxiv.org/abs/1605.07678.

[15]. J. Bankar and N. R. Gavai, "Convolutional Neural Network Based Inception V3 Model for Animal Classification," Int. J. Adv. Res. Comput. Commun. Eng., vol. 7, no. 5, pp. 142–146, 2018, doi: 10.17148/IJARCCE.2018.7529.

[16]. A. Lumini, L. Nanni, and G. Maguolo, "Deep learning for plankton and coral classification," Appl. Comput. Informatics, 2019, doi: 10.1016/j.aci.2019.11.004.

[17]. C. Wang et al., "Pulmonary image classification based on inception-v3 transfer learning model," IEEE Access, vol. 7, pp. 146533–146541, 2019, doi: 10.1109/ACCESS.2019.2946000.

[18]. Evan, M. Wulandari, and E. Syamsudin, "Recognition of pedestrian traffic light using tensorflow and SSD MobileNet V2," IOP Conf. Ser. Mater. Sci. Eng., vol. 1007, no. 1, 2020, doi: 10.1088/1757-899X/1007/1/012022.

[19]. S. A. Magalhães et al., "Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse," Sensors, vol. 21, no. 10, pp. 1–24, 2021, doi: 10.3390/s21103569.

[20]. K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification," Proc. - 2020 2nd Int. Conf. Inf. Technol. Comput. Appl. ITCA 2020, pp. 476–480, 2020, doi: 10.1109/ITCA52113.2020.00106.

[21]. W. Setiawan, "Perbandingan arsitektur convolutional neural network untuk klasifikasi fundus," SimanteC, vol. 7, no. 2, pp. 49–54, 2019.

[22]. M. A. Hossain and M. S. Alam Sajib, "Classification of Image using Convolutional Neural Network (CNN)," Glob. J. Comput. Sci. Technol., vol. 19, no. 2, pp. 13–18, 2019, doi: 10.34257/gjcstdvol19is2pg13.

[23]. N. E. Sahla, "A deep learning prediction model for object classification," Delft University of Technology, 2018.

[24]. A. Suhail, M. Jayabalan, and V. Thiruchelvam, "Convolutional neural network based object detection: A review," J. Crit. Rev., vol. 7, no. 11, pp. 786–792, 2020, doi: 10.31838/jcr.07.11.140.